Access Point Assignment Algorithms in WLANs Based on Throughput Objectives

Ioannis Koukoutsidis
Dept. Informatics and Telecommunications
National & Kapodistrian University of Athens
Ilissia, 157 84 Athens, Greece
Email: I.Koukoutsidis@di.uoa.gr

Vasilios A. Siris
Foundation for Research and Technology - Hellas
Institute of Computer Science (FORTH-ICS)
University of Crete
P.O. Box 1385, 711 10 Heraklion Crete, Greece
Email: vsiris@ics.forth.gr

***Abstract*—In this article we present branch-and-bound algorithms for the access point assignment problem in WLANs, when the objective function is based on the throughput of stations in the network. We consider: a) maximizing the aggregate throughput, b) achieving lexicographically max-min fair throughputs, c) achieving proportionally fair throughputs. The performance of all branch-and-bound algorithms is examined for various degrees of approximation. Thus we show trade-offs between the increased cost of exploration and improvement in the objective value. We further compare their performance to that of greedy algorithms, embedded as a depth-first-search in the branch-and-bound methods. An omnipresent result is the near-optimal performance of the greedy algorithms, which is particularly important when considering their practical application. In all cases, the performance of the algorithms improves as the distribution of wireless stations becomes more concentrated in areas of the network, as in hotspot topologies.**

***Index Terms*—WLAN, 802.11, access point assignment, throughput, optimization, branch-and-bound algorithms**

I. Introduction

In recent years, the need for increased coverage and capacity of WLANs has led to denser deployments of access points (APs). As a result, transmission ranges of APs are usually overlapping, and wireless stations (STAs) can be in communication range to several APs. The ensuing increased interference problems in the network can be alleviated to some extent by applying efficient power control or channel assignment mechanisms [1], [2]. For STAs that operate in the same frequency and are in communication range with several APs, the persisting problem is how to efficiently *assign* STAs to APs so as to improve the performance of the network.

In current WLANs, a STA scans the channel for beacon signals of nearby APs and connects to the AP which has the strongest Radio Signal Strength Indication (RSSI) value[1]. Based on the received power level, the physical layer transmission rate is selected together with the appropriate modulation and encoding scheme so that a given packet error rate is satisfied. In general, a higher receiver input power allows a higher transmission rate to be used. While this – herein called Max-RSSI – scheme ensures the best possible reception/transmission conditions, it does not consider the actual throughput of STAs, as occurring from the 802.11 contention mechanism. In other words, it does not explicitly consider performance as an objective.

Performance objectives we consider in this paper, associated with the throughput of wireless STAs are the following: i) maximizing the aggregate throughput ii) maximizing the minimum throughput, and iii) achieving proportionally fair throughputs. Each of these objectives has a different practical value. The first could be related to maximizing a linear utility function, usually defined from the point of view of a network operator. A well-known drawback of such an assignment is the induced unfairness in the throughput allocation. The second objective amounts to achieving as even as possible throughput values for all STAs. The third objective, based on the notion of proportional fairness introduced by Kelly [3], is known to achieve an efficient compromise between the first two, both increasing the aggregate throughput and maintaining fairness in the attained throughput values.

We consider a set of WLAN stations (STA) $\mathcal{S} = \{1, \dots, N\}$ and a set of access points (AP) $\mathcal{P} = \{1, \dots, P\}$, where $P < N$. Each STA j ($j = 1, \dots, N$) can be serviced by a subset of APs $\mathcal{P}_j$, which is called the *selection set* of STA j. The *quality* of servicing STA j by AP $i \in \mathcal{P}_j$ is denoted by $\theta_{j,i}$. In our case it represents the throughput of a STA, which depends on contention from other STAs in the same cell and the channel conditions.

For simplicity we shall use the throughput arising from *ideal random polling*:

$$\theta_{j,i} = \frac{1}{\sum_{u \in U_i} \frac{1}{r_{u,i}}} \ , \qquad (1)$$

where U_i is the set of STAs connected to AP i, and $r_{u,i}$ is the physical transmission rate of the connection of STA u with AP i, considered to be some function of SNR: $r_u = h(SNR(u,i))$.

This is a common simplification for the throughput of a STA (see [4]–[6]). It embodies two major aftereffects of the 802.11 mechanism. First, that the addition of more STAs in the same cell (i.e., connected to the same AP) always decreases the throughput of the existing STAs in the cell, due

This work has been supported by the NoE CONTENT (IST-84239).

[1]The RSSI is a unitless measure of the power of the signal, in an arbitrary scale.

WIOPT 2008, 1st – 3rd Apr 2008, Berlin, Germany.
 ISBN 978-963-9799-18-9
DOI 10.4108/ICST.WIOPT2008.3054

to increased contention. Secondly, that all STAs in the same cell obtain the same throughput, whose value is close to the minimum transmission rate of all STAs in the cell. This is a fundamental max-min fairness property of the protocol. The inverse of the right-hand-side of (1) is often termed the *load* of AP i [4], interpreted as the amount of time to transmit one unit of information by each of the STAs in the cell. The AP load can alternatively be used as performance objective in an optimization problem. Clearly, minimizing the sum of loads of all APs is equivalent to maximizing the aggregate throughput of all STAs. There is no such equivalence for the other objectives.

For each of the objectives, the problem is formulated as a nonlinear integer programming problem, with a non-convex and non-separable objective function. Analytical solutions were derived in [5] for the proportional fairness objective, in some simplified subcases and relaxations of the problem. In [6], the total load of all APs is treated as an energy function, which is minimized over time using a Gibbs sampler together with simulated annealing. In [4], minimizing the maximum load over all APs is considered. An approximate solution is obtained by solving the related fractional association problem (a linear programming problem), and then using a rounding method to approximate the optimal integral association.

In this paper, we consider the application of Branch-and-Bound (BB) algorithms for each of the aforementioned throughput objectives. In [5], a BB solution for the nonlinear integer program in the proportional fairness case was acquired by using a ready-made optimization software package (although no indices were given on the performance and efficiency of the algorithm). Here we take an inherently combinatorial approach. A key contribution is to identify appropriate bounding rules and branching procedures for each performance objective, that reduce the complexity of the search. We examine optimal as well as approximate solutions to the problem, i.e. considering a given relative error from the optimal. By construction, each BB algorithm has also an imbedded greedy sub-algorithm, and is guaranteed to find a solution at least as good as the greedy one. Based on this construction, we can show trade-offs between the cost of additional exploration of the state space and the increase in distance from the optimal, for various degrees of approximation.

We examine the performance of both BB and greedy algorithms for two characteristic topologies. In the first, STAs are uniformly distributed in the network area, while in the second STAs are concentrated in areas around each AP, thus creating "hotspots" in the network. We show that the performance of all algorithms is improved in such "hotspot topologies". Computationally efficient BB algorithms are constructed in the max-aggregate throughput and max-min fair throughput cases. In the proportional fairness case, the derived bounds are not tight enough to lead to a small computational cost. Nevertheless, in all cases the greedy algorithms are shown to be near-optimal; hence, taking into account their small computational complexity, they are more suitable for practical applications.

The rest of the paper is structured as follows. In Section II, we formally describe the objective functions used. Section III presents the construction of the BB and greedy algorithms for each throughput objective. In Section IV we present the performance of the algorithms for the two characteristic topologies. Section V is devoted to a discussion on the implementation of the algorithms. We end in Section VI with the major conclusions and further issues opened from this research.

II. Objective Functions

An assignment of STA j to AP i is denoted by a function $\alpha_j \stackrel{\text{def}}{=} \alpha(j) = i$. An assignment of STAs to APs is denoted by the set $\mathcal{A} = \{(1, \alpha_1), \ldots, (N, \alpha_N)\}$. To disburden notation, we denote this set simply as $\mathcal{A} = \{\alpha_1, \ldots, \alpha_N\}$. We emphasize the dependence of the quality of each connection on *all* other assignments by the identity $\theta_{j,i} = \theta_{j,i}(\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_N)$.

The objective is to find an optimal assignment $\mathcal{A}^* = (\alpha_1^*, \ldots, \alpha_N^*)$ such that

$$\mathcal{F}(\mathcal{A}^*) = \max_{\mathcal{A} \in \mathcal{R} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_N} \mathcal{F}(\mathcal{A}) \ ,$$

where the set of all possible assignments $\mathcal{R} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_N$ is the feasible set of the problem and $\mathcal{F}(\mathcal{A}) = \mathcal{F}(\alpha_1, \ldots, \alpha_N) \stackrel{\text{def}}{=} f(\theta_{1,\alpha_1}(\alpha_2, \ldots, \alpha_N), \ldots, \theta_{N,\alpha_N}(\alpha_1, \ldots, \alpha_{N-1}))$ is a function which can represent different aspects of throughput performance.

We consider sums of increasing utility functions of throughput, namely the *aggregate* throughput, $f(\Theta) = \sum_{j=1}^{N} \theta_{j,\alpha_j}$, and the sum of *proportionally fair* throughputs, $f(\Theta) = \sum_{j=1}^{N} \log \theta_{j,\alpha_j}$ [3].

Further, we consider the case of achieving *lexicographic max-min fairness*. This entails not only maximizing the minimum throughput, but finding the "most balanced" allocation of throughputs to STAs. It amounts to finding the lexicographically greatest throughput vector in the following sense: If $(\theta_{D,1}, \ldots, \theta_{D,N})$ is a vector of STAs throughputs in decreasing order, this is said to be lex-greater than another vector $(\theta'_{D,1}, \ldots, \theta'_{D,N})$, also sorted in decreasing order, if and only if $(\exists\ m > 0) : \ (\theta_{D,i} = \theta'_{D,i}\ \forall\ i < m) \wedge (\theta_{D,m} > \theta'_{D,m})$.

Hereafter in this paper, max-min fairness will be taken to mean lexicographic max-min fairness. The term lex-max-min fairness will also be used for shortness.

III. Branch-and-Bound Methods

We follow the formulation for a Branch-and-Bound approach for the generalized assignment problem in [7].

The BB algorithm will proceed by partially assigning STAs to APs. Let a *partial assignment*, when only STAs in the subset $\mathcal{S}' \subset \mathcal{S}$ have been assigned, be denoted by $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}} \stackrel{\text{def}}{=} \{(j, \alpha_j)\}_{j \in \mathcal{S}'}$. The subset of remaining possible assignments, when the associations of STAs in $\mathcal{S}'$ are held fixed, is $\mathcal{R}_{\mathcal{S}' \subset \mathcal{S}} = \{\mathcal{A} \in \mathcal{R} \mid \mathcal{A}_{\mathcal{S}' \subset \mathcal{S}} \subset \mathcal{A}\}$. Any assignment $\mathcal{A}_{\mathcal{S}'' \subseteq \mathcal{S}} \in \mathcal{R}_{\mathcal{S}' \subset \mathcal{S}}$, of STAs in a set $\mathcal{S}''$ such that $\mathcal{S}' \subset \mathcal{S}'' \subseteq S$, shall be called an *extended assignment* over the set $\mathcal{S}'$.

Consistently, we shall call an assignment where all STAs have been associated with an AP a *complete assignment*.

Each node in the constructed branching tree will be a partial or complete assignment, and at level k of the branching tree $k \leq N$ STAs will be assigned.

A. Bounding Rules

1) The Aggregate Throughput Case: We start with the following:

Lemma 1: Consider a partial assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$. Let $\mathcal{S}'_i$ be the subset of STAs assigned at this point to AP i ($i = 1, \ldots, P$), with $|\mathcal{S}'_i| = c_i$, and let assigned STA j have transmission rate $x_{j,i}$, $j = 1, \ldots, c_i$. If an unassigned STA with rate $x_{c_i+1,i}$ is to be connected to AP i, an increase in the objective function of that cell occurs if and only if

$$x_{c_i+1,i} > \frac{c_i}{\frac{1}{x_{1,i}} + \cdots + \frac{1}{x_{c_i,i}}} \ . \tag{2}$$

Proof: The condition follows straightforwardly from simple algebra. ■

That is, the new rate must be greater than the current aggregate throughput in order to have an improvement. From the necessary and sufficient condition the following corollary is obtained by induction.

Corollary 1: When STAs assigned sequentially to an AP have decreasing transmission rate values, the resulting sequence of aggregate throughput values is also decreasing.

An upper bound on the value of a partial assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$ can now be derived provided STAs assigned sequentially at each cell have decreasing transmission rate values. That is, for a partial assignment of c_i STAs to AP i, if a $(c_i + 1)$th assignment is made, we insist that $x_{c_i+1,i} \leq x_{c_i,i}$.

Theorem 1: For $M = \max\{r_{j,i},\ j \in \mathcal{S} \setminus \mathcal{S}',\ i \in \mathcal{P}_j\}$, an upper bound on the value of any complete assignment resulting from the partial assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$ is

$$U_{aggr}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) = \max_{i \in \mathcal{P}} \left((c_i + 1) \left(\frac{1}{\frac{1}{x_{1,i}} + \cdots + \frac{1}{x_{c_i,i}} + \frac{1}{M}} \right) + \sum_{k \neq i} \mathcal{F}_k \right) \ , \tag{3}$$

where $\mathcal{F}_k$ is the value of the objective function for cell k under the assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$.

Proof: The right part in (3) gives the maximum value of an extended assignment over $\mathcal{S}'$ with one more STA assigned. From Corollary 1, subsequent objective values in extended assignments will be decreasing. Hence the theorem is proved. ■

Remark 1: The additional cost for calculating the maximum transmission rate M of the remaining elements is linear in N, and it suffices to do an ordering of STAs' transmission rates to each AP only once.

2) The Max-Min Fairness Case: A well-known property of the average is that

$$\min_{j \in \mathcal{S}} \theta_{j,\alpha_j} \leq \frac{\sum_{j \in \mathcal{S}} \theta_{j,\alpha_j}}{N} \ . \tag{4}$$

Therefore, an upper bound on the value of a partial assignment for the max-min fairness case can be found from the corresponding upper bound for the aggregate throughput case:

$$U_{max-min}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) = \frac{U_{aggr}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}})}{N} \ . \tag{5}$$

This again holds provided STAs assigned sequentially to an AP have decreasing rate values.

3) The Proportional Fairness Case: The above analysis fails in the case of the proportional fairness objective, due to the non-linearity of the logarithmic function. A general upper bound for a partial assignment can be based on the fact that the addition of more STAs in a cell deteriorates throughput for all existing STAs in that cell.

Thus the following upper bound on the value of any complete assignment resulting from the partial assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$ is derived:

$$U_{pf}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) = \mathcal{F}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) + \sum_{s \notin \mathcal{S}'} \log \left(\max_{i \in \mathcal{P}_s} \theta^{\mathcal{S}'}_{s,i} \right) \ , \tag{6}$$

where $\mathcal{F}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}})$, is the sum of logarithms of the throughputs of STAs in $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$ and $\theta^{\mathcal{S}'}_{s,i}$ is the throughput of STA s in connection to AP i, given a system containing only the STAs in $\mathcal{S}'$.

B. Branching Procedures

The analysis in the aggregate and max-min fair throughput cases leads us to adhere to a monotonicity structure in our search, where subsequent assignments to the same AP have decreasing rate values. It is obvious that we can arrive at a complete enumeration of possible assignments of STAs to APs by keeping this monotonicity (since merely a re-arrangement of the order of these assignments would do the job).

Appropriately, we do not have to do anything else to accomplish this, other than select at each step the assignment with the highest upper bound on the objective function. (This follows from the strict monotonicity of the objective functions.) The assignment selected at level k of the branching tree is, in the aggregate throughput case,

$$(j, i) := \operatorname*{arg\,max}_{j \notin \mathcal{S}_k,\, i \in \mathcal{P}_j} \left(U_{aggr}(\mathcal{A}_k \cup (j, i)) \right) , \tag{7}$$

where $\mathcal{A}_k$, $\mathcal{S}_k$ contain the partial assignments and STAs assigned up to level k, respectively. Notice that this upper bound looks at the situation at the next two levels, and hence is tighter than the one derived by simply selecting the best $k + 1$ extended assignment.

In the max-min fairness case, we select the (STA, AP) pair which will yield the smallest deterioration in the current minimum throughput:

$$(j,i) := \arg\operatorname{lex}\max_{j \notin \mathcal{S}_k, i \in \mathcal{P}_j} (\min_{\substack{k \in \mathcal{S}_k \cup \{j\} \\ \ell \in \mathcal{P}_k}} \theta_{k,\ell}) \ . \tag{8}$$

To provide for lex-max-min fairness, we have included in the above operator the additional comparisons for the lexicographically greatest throughput vector.

Remark 2: Note that a lex-max-min ordering of throughputs corresponds to a lex-min-max ordering of loads, but not vice versa. Since many STAs may be associated to an AP, the same lex-min-max load vector may occur for unequal throughput vectors (an example is shown in [4]). Hence, to achieve the "most balanced" throughput association, optimization must necessarily consider a throughput objective.

In the proportional fairness case the assignment which yields the highest upper bound $U_{pf}(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}})$ to an extended assignment will be selected:

$$(j,i) := \arg\max_{j \notin \mathcal{S}_k,\, i \in \mathcal{P}_j} \left(U_{pf}(\mathcal{A}_k \cup (j,i)) \right) \ . \tag{9}$$

Based on the upper bound expressions (3),(5),(6), node elimination will be performed as follows. Let the set $\mathcal{CA}$ contain all complete assignments derived at some point of the execution of the algorithm, and denote by $\hat{\mathcal{F}}$ the maximum value of any assignment in the set, $\hat{\mathcal{F}} = \max_{\mathcal{A} \in \mathcal{CA}} \mathcal{F}(\mathcal{A})$. If for some partial assignment $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$, $U(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) < \hat{\mathcal{F}}$, then we no more need to check branches from this node and this node, or any already derived branches from this node are eliminated.

1) Recursive Branching: All algorithms combine Branch-and-Bound with a recursive technique akin to dynamic programming, that we call *recursive branching*. It arrives without delay at the "most promising" solution and then backtracks at previous levels trying to improve this solution. Such an approach was followed in [8] for the problem of assigning workforce on different units of a production line. Key elements of the algorithm are the following:

1) Initially, at levels $k = 0, \ldots, N-1$ we assign STAs to APs sequentially in a depth-first manner, so that we arrive at a complete assignment after N steps. At each step, the assignment with the highest upper bound on the objective function is selected, as in (7), (8), or (9).
2) After we arrive at the "most likely" solution, we backtrack at previous levels in search for improved alternatives. A dynamic programming technique is applied: we ensure that at any level k, $k = 1, \ldots, N-1$, given $\mathcal{A}_k$, the sequence of remaining assignments is optimal. (The algorithm backtracks from level k to level $k-1$ when the remaining $N-k+1$ STAs have been optimally assigned.) Examined level-k assignments are stored in a set $\mathcal{E}_k$; these are excluded while searching for improved assignments at levels greater or equal to k (i.e., until we backtrack to level $k-1$). After we have reached level 0 again, the assignment of STAs is optimal and the algorithm stops.

The algorithm is easy to program and requires a small storage space. The recursive algorithm is shown in pseudocode as Algorithm 1. To provide a compact pseudocode for all

Algorithm 1 Recursive Branch-and-Bound algorithm

1: $\mathcal{S}_0 := \emptyset$; $\mathcal{A}_0 := \emptyset$;
2: $\mathcal{E}_k := \emptyset$, $k = 1, \ldots, N$;{$\mathcal{E}_k$ stores assignments excluded from levels $\geq k$}
3: k:=1; $\hat{\mathcal{F}} := -\infty$; {$k$: current level}
4: **while** $k > 0$ **do**
5: read $\mathcal{A}_k$, $\mathcal{S}_k$;
6: select (j,i) according to (7), (8) or (9), where $(j,i) \notin \mathcal{E}_\ell$, $\ell = 1 \ldots, k$;
7: **if** $(U(\mathcal{A}_k \cup (j,i))) < \hat{\mathcal{F}}$ **or** (j,i) =NULL **then**
8: $\mathcal{E}_k := \emptyset$; $k := k-1$;
9: **else**
10: $\mathcal{S}_k := \mathcal{S}_{k-1} \cup \{j\}$; $\mathcal{A}_k := \mathcal{A}_{k-1} \cup (j,i)$;
11: $\mathcal{E}_k := \mathcal{E}_k \cup (j,i)$; $k := k+1$;
12: **end if**
13: **if** $k = N+1$ **then**
14: $\hat{\mathcal{F}} := \mathcal{F}(\mathcal{A}_{k-1})$; $\mathcal{A}^* := \mathcal{A}_{k-1}$;
15: $\mathcal{E}_{k-1} := \emptyset$; $k := k-2$;
16: **end if**
17: **end while**

objective functions, we have dropped subscripts on upper bound functions. We also note that in the max-min fairness case an extra step is required that checks for the lexicographically greatest vector at level $k = N$. Additionally in our implementation, we have added an extra control on line 7: backtracking to previous levels is also realized if the maximum throughput value in an examined assignment is smaller than the maximum throughput of the current best assignment. In case of ties, a node or unassigned STA for branching can also be selected randomly among equal-valued alternatives. Finally, in case where some STAs can only be assigned to a single AP, the starting node contains these assignments.

C. Approximate Solutions

Suppose we have a complete assignment with value $\hat{\mathcal{F}}$, and U is the maximum upper bound over all active (i.e., non-eliminated) nodes. For a given relative error σ, this complete assignment can be adopted as an approximate solution when $\frac{U - \hat{\mathcal{F}}}{U} \leq \sigma$. In our recursive algorithm, we eliminate during the search all partial assignments $\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}$ for which

$$\frac{U(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}}) - \hat{\mathcal{F}}}{U(\mathcal{A}_{\mathcal{S}' \subset \mathcal{S}})} \leq \sigma \ . \tag{10}$$

D. Greedy Algorithms

In all throughput objective cases, the depth-first-search part of the BB algorithm constitutes a *greedy* algorithm. The algorithms are characterized as greedy because they attempt to choose the best possible next assignment pair based on the current assignment and some local optimization criterion. In our case, the optimization criterion consists of maximizing the upper bound of a partial assignment. This allows to look more ahead into the future, and hence improve performance. All algorithms are *adaptive*, in the sense that there is no a priori fixed sequence of assignments, but each new decision is made based on the previous decisions.

IV. Examples

In this section we present some examples with two realistic topologies in an 802.11a network. We distribute 10 STAs in a $100\,\mathrm{m}\times 100\,\mathrm{m}$ square area with 3 APs fixed at coordinates (20,20), (50,50), (80,80). In the first topology STAs are uniformly distributed in the area, while in the second multiple hotspot areas are created.

TABLE I
Power-dependent data rates in 802.11a [9] (for a 1000 bytes PSDU and OFDM modulation at 5 GHz)

Data rate (Mbps)	Minimum sensitivity (dBm)
6	-82
9	-81
12	-79
18	-77
24	-74
36	-70
48	-66
54	-65

Physical transmission rates depending on received input power levels are shown in Table I. Received powers in dBm are calculated as

$$P_r = P_t - (P_{ref} + 10\gamma \log_{10} d) \ , \qquad (11)$$

where P_t (in dBm) is the transmit power, typically 20 dBm (100 mW), P_{ref} is the reference loss at a distance of 1m from the transmitter, typically 46.4 dB at 5 GHz, γ is the path loss exponent set at 2.7 (this choice was made for an intermediate environment, between 2.0 for open space and 3.5 for an office building environment), and d the distance from the transmitter in meters.

To evaluate the performance of the algorithms, we mainly compare them to an exhaustive search. The time complexity of the algorithms in all cases is comprised of two costs: the cost for calculating the objective value of an assignment, or an upper bound to the objective value for a partial assignment, and the cost for the selection algorithm which subsequently finds the optimal assignment. Calculating the upper bound of a partial assignment has an increased cost, mainly due to additional selection operations for finding maximal objects. However, as was explained in Remark 1, and given that the number of APs is usually small, the additional complexity is not extravagant.

To simplify our evaluation, we only consider the total number of function value comparisons (objective values or upper bounds to objective values in case of partial assignments) – and vector comparisons in the max-min fairness case – as a measure of the complexity of the algorithms. This is contrasted to the respective cost incurred by a selection algorithm in an exhaustive search, which in the worst case is equal to P^N.

It is worth noting that the demonstrated performance corresponds to minimum memory requirements. The number of comparisons can be further reduced, e.g. by storing the values of partial assignments, so that re-computations are avoided when the algorithm backtracks or goes forward in levels of the branching tree. Such programming devices are not investigated in this paper. It should be mentioned that, in the results, assignments are optimal in a maximal sense. Due to the discrete set of data rate values, different assignments may lead to the same objective value.

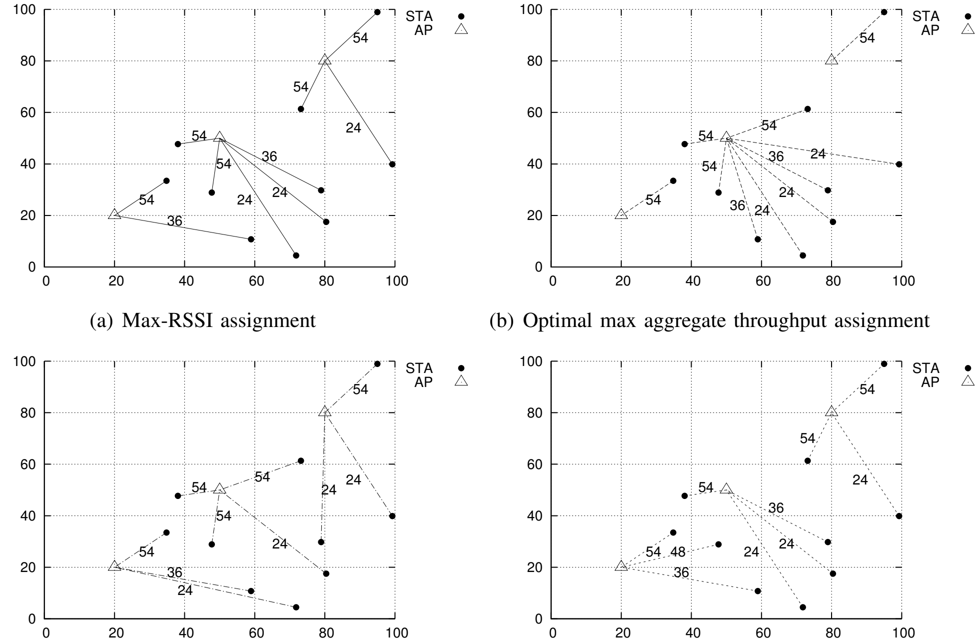
Example 1. Uniform Topology: Intuitively, the case of a uniform topology is the hardest for the assignment problem, since STAs may have similar signal quality to several APs. Optimal assignments for a selected instance of such a topology are shown in Fig.1, where physical transmission rates are shown for each link (in Mbps). Average results for the performance of the BB and greedy algorithms for each throughput objective in such a topology are presented in Table II, for 30 randomly generated instances of the topology. For each objective the respective BB algorithms are examined for different values of the relative approximation error σ. As σ goes to zero, the algorithms converge towards the optimal assignment. In all cases an approximate algorithm initially performs the same depth-first search as the greedy algorithm, therefore its performance is at least good as the one of the greedy algorithm. Further, the same sequence of pseudo-random instances is examined so that results are directly comparable.

In the results, the second row of each sub-table shows the percentage of cases where the greedy algorithm reached the optimal assignment, and the third the relative error compared to the objective value of the optimal assignment, averaged over all experiments. In the max-min throughput case, the relative error is measured based on the minimum throughput values.[2]

The best behavior is exhibited by the max-aggregate-throughput BB algorithm, outperforming exhaustive search. In the lex-max-min fairness case, the BB method carries out a smaller number of comparisons for an approximation error of about 10%. In the proportional fairness case, the BB method does not possess bounds tight enough to end in a small number of steps. Nevertheless, it is important to notice that greedy algorithms for all objective functions perform well.[3] The average relative error is smaller than 3% in Tables II(a),II(c), and goes up to 12.19% in Table II(b), which can be considered acceptable. In fact, making the algorithms "more optimal" by reducing the approximation error σ only yields a very small advantage compared to the cost of performing additional comparisons. In many cases additional exploration

[2]The approximation error for the BB algorithm in the max-min fairness case is based on the upper bound of the minimum throughput. This is different from the average relative error calculated in the third row of Table II(b), which turns out to be a little higher.

[3]The increased number of comparisons for the greedy algorithm in the lex-max-min fairness case as opposed to the aggregate throughput and proportionally fair throughput cases is due to additional comparisons for lex-greater throughput vectors.



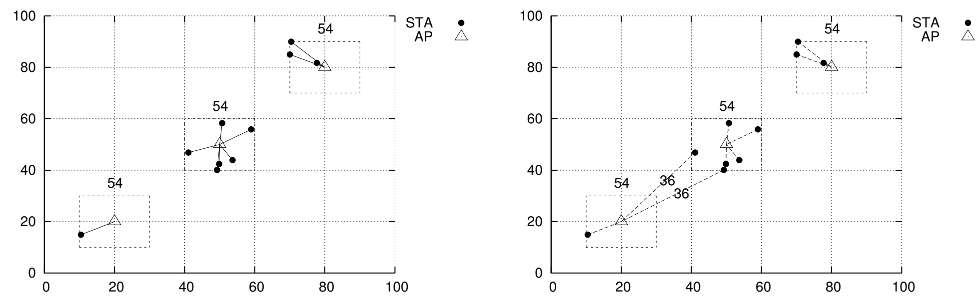
(a) Max-RSSI assignment

(b) Optimal max aggregate throughput assignment

(c) Optimal max-min throughput assignment

(d) Optimal proportionally fair throughput assignment

Fig. 1. Assignments of STAs to APs in a selected instance of the uniform topology.



(a) Max-RSSI and optimal max aggregate throughput assignment

(b) Optimal max-min fair and proportionally fair throughput assignment

Fig. 2. Assignments of STAs to APs in a selected instance of the hotspot topology.

does not manage to find a better solution that the greedy algorithm, and hence such greedy algorithms with very small computational cost may be sufficient in practical situations.

It is also shown that in the case of a uniform topology, the current Max-RSSI assignment scheme does not reach any of the throughput objectives, and performs significantly worse than the greedy algorithms we have considered.

Example 2: Hotspot Topology: Hotspots are considered within a 20m $\times$ 20m area around each AP. STAs are placed with probability 0.5 at the central hotspot, and at each of the remaining hotspots with probability 0.25.

Optimal assignments for a selected instance of such a topology are shown graphically in Fig. 2. Hotspot areas are shown in dashed boxes; in each area STAs are connected at the maximum rate of 54 Mbps, shown on top of each box. Transmission rates are also shown for links outside these boxes. For the example shown in Fig. 2 the solutions to the max-aggregate throughput and Max-RSSI assignments coincide, as well as the solutions for the optimal max-min and proportionally fair assignments. This is frequently the case in such a topology.

The structure of a hotspot topology makes the combinatorial assignment problem much easier, since, for each STA, there are greater disparities between connection rates to each AP, and hence good or bad decisions are distinguished more easily. This reflects on the performance of the BB as well as greedy algorithms, which are much more improved in such a case. We have again conducted experiments over 30 random instances of this topology, and results for the average performance of the BB and greedy algorithms are shown in Table III.

It is striking that for the max aggregate throughput case the greedy algorithm always converged to the optimal assign-

TABLE II

AVERAGE PERFORMANCE OF BB (WITH VARIOUS DEGREES OF APPROXIMATION) AND GREEDY ALGORITHMS FOR DIFFERENT THROUGHPUT OBJECTIVES, IN THE UNIFORM TOPOLOGY

(a) Max aggregate throughput

	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.1$	Greedy	Max-RSSI
Avg. number of comparisons	52456	33839	15157	12740	165	N/A
% of optimality cases	100	66.67	53.33	46.67	46.67	0
Avg. relative error (%)	0	0.74	1.88	2.28	2.41	10.02

(b) Max-min throughput

	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.1$	Greedy	Max-RSSI
Avg. number of comparisons	179170	164830	107578	48662	184	N/A
% of optimality cases	100	80	43.33	36.67	36.67	10
Avg. relative error (%)	0	1.67	5.07	10.31	12.19	26.68

(c) Proportionally fair throughputs

	$\sigma = 0.15$	$\sigma = 0.2$	$\sigma = 0.25$	Greedy	Max-RSSI
Avg. number of comparisons	379133	252516	31065	165	N/A
% of optimality cases	36.67	33.33	33.33	30	6.67
Avg. relative error (%)	0.87	0.96	0.96	1.08	6.54

ment. Conformingly the BB algorithm also had very good performance and found the optimal assignment in a single depth-first search (with a higher total number of comparisons). We also remark that for the hotspot topology, the max-min fair BB algorithm has significantly improved performance. Even for $\sigma = 0$, the number of comparisons is much smaller than in the exhaustive search case. Improved performance is maintained when allowing a higher approximation error, as well as for the greedy algorithm. On the other hand, for a reasonable approximation error the proportionally fair BB algorithm does not exhibit good behavior, even for a hotspot topology. However, the performance of the greedy algorithm is also much improved compared to the uniform topology case, showing a high percentage of optimality cases and an extremely small average relative error.

Finally, as it is anticipated, in a hotspot topology case the Max-RSSI scheme is a near-optimal heuristic for maximizing the aggregate throughput. However, it is inappropriate for the remaining objectives.

V. IMPLEMENTATION ISSUES

The feasible transmission rate of each STA to each AP in its range is needed as input to the algorithms. The necessary mechanisms for this task are already implemented in WLANs for supporting the current RSSI scheme. The transmission rate information must be delivered to a control center in the network which will carry out the AP assignment task. In current WLANs with a few tens or hundred nodes, the centralized gathering and processing of information should be feasible in terms of the processing cost and delays (wireline connections are assumed between APs). For even larger networks, a practical approach is to partition the network and making the assignment decisions separately for each partition. The optimality loss by making assignments separately for each partition is likely to be negligible.

An important issue in the network is churn, with STAs terminating or initiating new connections randomly. Additional dynamic effects caused by changes of STAs' positions and varying channel conditions are less important in WLANs, which are characterized by low mobility. It is evident that, depending on the rate of changes in the topology, a balance must be sought between the cost of successive re-assignments and the distance from the optimal assignment, for the selected performance objective. Given their witnessed good approximation to optimal assignments in all cases, the greedy algorithms having very low complexity can be run in a practical online scheme.

Consider a number of STAs in a WLAN, optimally assigned to APs at a certain time instant. At a subsequent instant some STAs have left the network and some new ones have joined in, connecting to APs according to some non-optimal criterion (e.g., Max-RSSI). It is hard to calculate the distance of the new assignment from the optimal assignment in such a case. Even the assignment of the remaining initial STAs may not be optimal. A practical online scheme consists of running the greedy algorithm for the network with the new topology. This will give a good approximation to the new optimal assignment. Then we can compare it with the value of the new assignment,

TABLE III
AVERAGE PERFORMANCE OF BB (WITH VARIOUS DEGREES OF APPROXIMATION) AND GREEDY ALGORITHMS FOR DIFFERENT THROUGHPUT OBJECTIVES, IN THE HOTSPOT TOPOLOGY

(a) Max aggregate throughput

	$\sigma = 0$	Greedy	Max-RSSI
Avg. number of comparisons	318	165	N/A
% of optimality cases	100	100	83.33
Avg. relative error (%)	0	0	4.17

(b) Max-min throughput

	$\sigma = 0$	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.1$	Greedy	Max-RSSI
Avg. number of comparisons	23164	23090	2529	380	198	N/A
% of optimality cases	100	100	93.33	83.33	83.33	6.67
Avg. relative error (%)	0	0	0.21	0.89	0.89	30.31

(c) Proportionally fair throughputs

	$\sigma = 0.15$	$\sigma = 0.2$	$\sigma = 0.25$	Greedy	Max-RSSI
Avg. number of comparisons	220232	46529	8112	165	N/A
% of optimality cases	70	66.67	66.67	66.67	6.67
Avg. relative error (%)	0.27	0.36	0.36	0.36	4.2

and decide to do a re-assignment if the difference exceeds some threshold value.

VI. CONCLUSIONS AND FURTHER DISCUSSION

In this paper, we investigated the application of BB algorithms for optimally assigning STAs to APs based on several throughput objectives. We arrived at computationally efficient algorithms in the max-aggregate throughput and lex-max-min fair throughput cases, with extremely good performance in hotspot topologies. For the proportional fairness objective, simple tight bounds could not be derived and an analytical BB algorithm based on the nonlinear integer program formulation may prove to be more efficient. Apart from that, the focus was on studying approximate algorithms in all cases. We constructed suboptimal greedy algorithms based on the same bounds used in the BB techniques. These showed to behave remarkably well in all cases, being only at a small relative distance from the optimal and arriving at the exact optimal assignment for a large percentage of our tests. Their performance further improved when considering hotspot topologies. On the contrary, it was observed that the current Max-RSSI assignment scheme which does not target performance explicitly is only a good heuristic for maximizing the aggregate throughput in a hotspot topology (yet inferior than the corresponding greedy algorithm).

The observed good performance of the greedy algorithms is very significant from an engineering perspective, since a low-cost good algorithm is all we need in a real system. We searched for properties of these algorithms and the problem itself that would account for this very good performance. As we said, favorable properties are that the algorithms are adaptive, always searching for the current best pair in each subsequent assignment, and that the local optimization criterion (based on maximizing the upper bound of the partial assignment) allows to look more ahead into the future. It is also worth adding that each considered assignment problem is the intersection of easy-to-solve subproblems (one for each AP) that have a matroidal structure. However, apart from these observations, no solid property was found. It is of course an open problem to search for approximation bounds for these algorithms.

There is also the question of how the algorithms perform when the scale of the problem increases. That is, either the number of STAs and/or APs increases, or there is a larger geographical area. Arguably, this depends more on the structure of the problem, and how uniform transmission rates to different APs are. We have seen that the performance of the algorithms improves when the distribution of STAs is more concentrated in hotspot cells. Thus, if such a structure is kept, we would expect the algorithms to perform well when the scale increases. Preliminary tests we conducted confirmed this, and showed the BB algorithms to achieve significantly improved performance over exhaustive search, in the same cases as with the smaller scale. However, in some random instances the computation time was again extravagant, which inhibited us from presenting average results for larger scales. We advocate that again, in a real scenario and for larger scales, the greedy algorithms should be employed.

Another issue concerns the construction of *online* algorithms. In the previous section we discussed an online scheme for AP assignment, however it remains that the algorithms are inherently *offline*. An online algorithm takes as input the sequence of newly arriving STAs, and outputs for each the best possible association, without considering global re-assignments. It is interesting to apply greedy schemes here and evaluate their performance with respect to the offline greedy algorithms as well as the optimal assignments.

Finally, it is worth discussing an extension of the assignment problem, which we call the *incomplete coverage problem*. Consider a modified version of the problem where we allow that only a subset of $N-k$ STAs, where $0 < k < N$, be serviced, and we would like to determine the best assignment of $N-k$ STAs to APs, so that the objective function of the system is optimized.

A practical instance in which it appears is when some

STAs are very distant from any AP or generally have bad channel conditions, so that their inclusion in the network would largely deteriorate performance of the whole system and it is preferable to exclude them. The problem is also raised in admission control, since with each connection request of a new STA one may need to decide on a number of STAs to evict in order to maintain QoS.

Note that exhaustive search in this case requires $\binom{N}{N-k} \cdot P^{N-k}$ comparisons. The same branch and bound algorithm can be used to solve the problems, with the difference that a complete assignment is now supposed to occur at the $(N-k)$th level. However, it was experimentally confirmed this is a bad approach to take, both because of the increase in state space and because bounds in this case are much less tight. The greedy algorithms for each objective can instead be applied with the new stopping criterion.

ACKNOWLEDGEMENT

We wish to thank Vaggelis Angelakis from FORTH-ICS for participating in helpful discussions at the beginning of this work.

REFERENCES

[1] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 3, pp. 19–31, 2005.

[2] P. V. Bahl, M. T. Hajiaghayi, K. Jain, S. V. Mirrokni, L. Qiu, and A. Saberi, "Cell breathing in Wireless LANs: Algorithms and evaluation," *IEEE Trans. Mobile Computing*, vol. 6, no. 2, pp. 164–178, 2007.

[3] F. Kelly, "Charging and rate control for elastic traffic," *Europ. Trans. Telecommun.*, vol. 8, pp. 33–37, 1997, a corrected version of the paper appears online at http://www.statslab. cam.ac.uk/~frank/elastic.ps.

[4] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancing in Wireless LANs using association control," in *Proc. MobiCom '04*. New York, NY, USA: ACM Press, 2004, pp. 315–329.

[5] A. Kumar and V. Kumar, "Optimal association of stations and APs in an IEEE 802.11 WLAN," in *Proc. National Conference on Communications (NCC)*, IIT Kharagpur, India, Jan./Feb. 2005.

[6] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks." in *Proc. Infocom*, Anchorage, Alaska, USA, 2007, pp. 1451–1459.

[7] G. Korablev and E. Andreichuk, "Generalization of the assignment problem," *Cybernetics and Systems Analysis*, vol. 13, no. 4, pp. 561–565, 1977, translated from Kibernetika, 4, 84-87, July-August, 1977.

[8] É. Volynskii, A. Vyatkin, and L. Krasin, "An assignment problem," *Cybernetics and Systems Analysis*, vol. 9, no. 2, pp. 317–320, 1973, translated from Kibernetika, 2, 111-113, March-April, 1973.

[9] *IEEE Std 802.11a-1999 (R2003)*, IEEE Computer Society, 2003.